UML-oriented Risk Analysis in Manufacturing Systems

J.Jirsa, J.Žáček

Abstract

Whenever we want to avoid failures or hazardous events in today's complex technological systems, it is advisable to carry out appropriate risk management. One of the most important aspects of risk management is the risk analysis process. The aim of this paper is to show a new risk analysis method based on the Unified Modelling Language (UML), which is successfully used in software engineering for describing the problem domain. The paper also includes a small practical example. It also shows a new risk analysis method based on an example of an unreeling process in cable manufacturing.

Keywords: risk analysis, Unified Modelling Language, manufacturing systems.

1 Introduction

Risk analysis and risk control have been very frequently used words in recent years. It seems to be a modern trend to speak about risk, mainly in economics or in management of the environment. However if we focus on our everyday lives, we will find that there are many situations when we subconsciously make a risk analysis.

We do not usually recognise particular phases of our intuitive risk analysis, and we cannot apply this biological process directly in technical applications. The reason is very simple: we might omit some important factors during our intuitive risk analysis which can lead to fatal consequences. Present-day technologies are complex systems, and they work with various materials and utilise demanding processes. During the life-cycle of these technologies, hazardous events or failures can occur. So we try to find ways to prevent all potential damage. For this we need risk analysis.

The paper focuses primarily on risk analysis for technological systems. Our aim is to present a new application of standard software modelling tools to improve and enrich computer processing, and to simplify the steps in risk analysis. The paper therefore provides a new interdisciplinary view of the risk analysis issue, as will be discussed below.

2 Risk analysis process

As mentioned above, each of us applies risk analysis several times per day. We do this process fully automatically (subconsciously). We usually do not think about it in greater detail. For example, as we leave home, we try to remember if all electric, gas and water devices have been switched off or closed, including the kitchen stove and the water taps. A similar example is when we prepare for our holiday: we think about possible hazards and we try to avoid them or at least to be well prepared. It is obvious that the same sort of thinking is applicable in technical branches. Let us take a look at three common characteristic questions, mentioned by Tichý in [1]:

1. What failures can occur in the inspected object or process?
2. How often can these failures arise?
3. What will happen after the failure occurs?

These questions are universal enough, and can be applied to every human activity. However, for real usage, especially in technical applications, it is necessary to specify concrete steps with specific rules. It is recommended to follow the general risk analysis process for technological systems defined in the IEC 300-3-9 standard [2]. In particular, it is necessary to take the following steps:

1. define the scope of the analysis,
2. identify the hazard and make an initial evaluation of the consequences,
3. estimate the risk,
4. verify,
5. make documentation,
6. update the analysis.

As might be expected, each step can be subdivided into more detailed tasks. Although these steps may appear simple and easy, it is often quite difficult to implement them.

3 Risk analysis techniques

In the course of history, many techniques have been developed for making a risk analysis, especially in the last hundred years. It is necessary to recognise that this is an ongoing process. We can observe the progress in risk analysis techniques in response to rapid technical advances. Some modern techniques are derived from older methods, which have been

Table 1: The most widely used risk analysis methods

Method	**Suitable for complex systems**	**Suitable for new systems**	**Quantitative analysis**	**Top-down or bottom-up**
Event Tree Analysis (ETA)	no	no	yes	bottom-up
Failure Modes and Effects Analysis (FMEA)	no	no	yes	bottom-up
Fault Tree Analysis (FTA)	yes	yes	yes	top-down
Hazard and Operability Studies (HAZOP)	yes	yes	no	bottom-up
Human Reliability Analysis (HRA)	yes	yes	yes	bottom-up
Reliability Block Diagrams (RBD)	no	no	yes	top-down
Preliminary Hazard Analysis (PHA)	yes	yes	no	bottom-up

suitably modified and enriched to fulfil current needs and make use of new possibilities. However, risk analysis techniques have originated in various fields of activity and in various historical eras (in the scope of technical and technological invention), but all of them use the rules of systematic analysis and logic. We can see the application of two main logical principles: *induction* and *deduction*. Induction is used when we investigate possible consequences of hazardous events. On the other hand, deduction is applied to find out possible causes of hazards or failure modes. In terms of risk analysis, these principles are called *bottom-up* for induction, and *top-down* for deduction. Of course there are ways to combine these two principles, but we can also apply them separately. In addition, risk analysis methods can be divided from another point of view, as can be seen in [4].

The qualitative or quantitative character (or both) of each method will now be discussed. This distinction is based on whether the method provides numerical results. Another aspect of risk analysis methods is the output format. This issue is often determined by the principles that are applied, the structure of the system (or process), and by the accompanying effort to achieve clear visualisation. Thus we can find verbal, tabular or graphical outputs. For a complex technological system with many parts spread over a wide area, it can be a very hard task to describe this system in purely textual form, and it may be better to choose a suitable graphical representation. The most widely used risk analysis methods are presented in Table 1, which has been taken from [4, 2] and reduced.

All methods mentioned here strictly use logical principles. However, let us focus on an alternative way of making an analysis, which is widely used in other technical branches: software development.

4 Object-oriented principles

At the beginning of each software project, it is necessary to make several analytical steps. In these steps, software developers attempt to identify and describe all desirable entities, their relationships and their behaviour. Here we can clearly see a basic similarity with the risk analysis process: the initial steps are the same — see section 2.

During the last three decades, object-oriented approaches have often been used for these purposes. Object-oriented approaches are based on the idea that the world consists of objects which interact with each other. Objects are characterised by the following features:

- encapsulation
- inheritance
- polymorphism

The term "encapsulation" indicates that the attributes and functions of the entity are joined together into one specific object. Attributes describe the state of the object, and functions can change the state and behaviour of the object. Inheritance reflects the everyday reality of the evolution of objects. It allows a new object to be derived from one or more parent objects, and during inheritance some new features can also be appended. Polymorphism shows the behaviour which different object types have in common.

The idea of object-oriented approaches also involves structural aspects, so we can consider basic

relationships, such as *aggregation, association* and *composition*.

Interaction between objects is provided by sending messages. The source object sends a request for some function on the target object. The target will react to the incoming event in accordance with its state and conditions.

The same principle has been unconsciously applied in technically-oriented risk analysis for a long time. We may inspect an object (e.g. a manufacturing system, an engine, a component) in a view of its properties, functions and interconnections with other objects. In general, we can observe specific hazards assigned to a specific object. Therefore we can say that these hazards are encapsulated in the object. Our experience with similar objects gives us a guideline for estimating the potential hazard, so we work with inheritance. Polymorphism in risk analysis can be seen in the following way: the same hazard can be caused by several different objects.

The Unified Modelling Language was developed for graphical visualisation of previous concepts, and also fits well for several other purposes.

5 UML modelling tools

Unified Modelling Language is a specific set of tools which can help in several fields of activity, not only in software engineering. It uses the object-oriented approach mentioned above and adds some complementary tools for a better description of the structure and behaviour of the system. We demonstrate that all these features can be utilised in several stages of risk management, mainly during a description of the system (or process) and also in visual hazard scenario modelling. Although UML is a very complex language, let us take a brief look into its composition. The language includes the following basic construction blocks [3]:

- subjects
- relationships
- diagrams

Subjects (or abstractions) can be further subdivided to:

- structural abstractions — nouns e.g. classes, interfaces, collaboration, use cases, etc.
- behaviour — modal verbs, e.g. interaction, state;
- aggregations — packages for grouping significantly related components;
- comments — additional useful annotations extending the model.

Diagrams are a graphical representation of the model. There are symbols with predefined syntax and semantics to show the model in several views. Obviously, diagrams allow better orientation in a description of the system, especially when several people are participating in the project. In this case, diagrams are an unambiguous form of description used in team cooperation, and they are often more effective than huge paragraphs of text. An important difference between models and diagrams is that when we remove a symbol from a diagram, it does not mean that the corresponding parts in the model are automatically discarded.

All descriptions, recommendations and simple examples can be found in the UML specification [9]. We can also see an interesting UML feature: the metamodel approach. This means that a language definition can be described by its own means.

6 Risk analysis based on UML

In this part of our paper we discuss the use of UML as a helpful tool for risk assessment. We have tried above to briefly describe the basic principles of the risk analysis process and an object-oriented approach used in software application design. Although there are many similarities between these different processes (both are usually organised as a project [6]), it is not easy to apply the steps used in software design to a risk analysis. UML does not provide recommended methodologies for using its own tools and the sequence in which to make the UML parts. Several methodologies have been developed in software engineering that aim to describe the right order, for example RUP (Rational Unified Process), see [7]. We can also take some inspiration from these methodologies, but the differences in our domain should not be forgotten. Let us note that the RUP methodology also includes some steps related to risk assessment. An important consideration is that the whole process of risk analysis cannot be finished at once and cannot be done quickly. If we want to obtain appropriate and valuable results, we have to iterate as many times as necessary. We would now like to propose a new UML-based method for risk assessment. This new method consists of the following steps:

1. describe the system structure by *Class diagram* or *Component diagram*,
2. describe the behaviour by *Activity, Sequence* or *State diagrams*,
3. identify, qualify and add potential hazards into *Class diagrams*,
4. create rough hazard scenarios by *Use Case diagrams*,
5. describe detailed risk scenarios with the use of *Interaction* or *Activity diagrams*,
6. evaluate the results and the documentation.

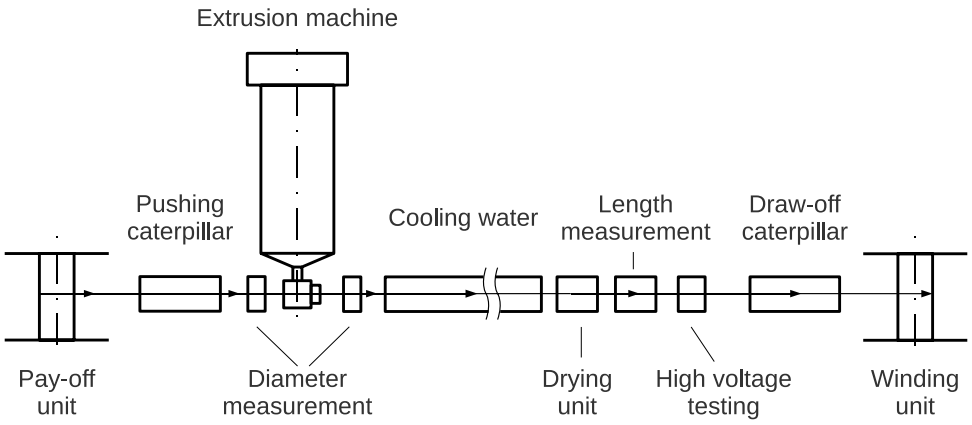

Fig. 1: NFA2X cable core processing line

7 Practical example

Let us take a look at a small practical example from electro-technical manufacturing. We will be investigating possible risks in a processing line that produces NFA2X cable core. This is an aluminium twisted wire core with XLPE (cross-linked polyethylene) insulation. A simple scheme of the processing line topology is shown in Fig. 1, sketching its real configuration at PRAKAB Pražská kabelovna, a.s. company.

The drum (or reel) with the wire core is fastened in the portal pay-off unit. The wire core is reeled off by the pushing caterpillar and goes into the head of the extrusion machine, where the insulation is deposited. In the next step, a new cable core is drawn through the water-filled cooling trough. Then it is dried up by the compressed air, tested for dielectric strength by high voltage and wound up to the drum. Appropriate tension of the wire core is provided by the draw caterpillar placed in front of the portal winding unit. There are also two diameter monitoring devices and a length measurement device.

As the risk analysis process of the whole line is very large, we will focus only on the first device in the line – the portal pay-off unit. We limit our analysis in order to focus on presenting UML as a risk analysis support tool. For the same reason, we will not work out the quantitative part. Our aim is to identify potential hazards that could threaten the smoothness of the operation and to show possible realisation scenarios. If, as in most cases, we need a quantification, we can make the Risk Priority Number (RPN) [8] calculation used in Failure Mode and Effects Analysis (FMEA). The results are given by equation (1)

$$RPN = Sv \times Lk \times Dt \tag{1}$$

where the Sv is a *severity* value, Lk is a *likelihood* value and Dt means *detection*. All three values are estimated and assigned during risk analysis, usually from a predefined degree scale. It is important to remember that the degree scale should not begin with zero, and should have a suitable range.

First, we should start with object identification. There are three major objects that participate in the unwinding process:

- the portal pay-off unit,
- the cable reel,
- the operating staff.

Let us focus on the first of these. Fig. 2 shows the design of the portal pay-off unit. For better understanding, a cable reel is also drawn in its working position, but not fastened up. We can clearly see the structure and the relationships between the components. The portal consists of two movable interconnected arms. Each arm has a lifter on its pole, which is coupled to the other on the opposite side. One arm is equipped with a compressed-air brake to supply reel braking. All movements are controlled by electrical drives. The whole unit traverses on floor rails across the unwinding direction. This feature is necessary for correct unwinding, otherwise the cable core can be damaged by the reel fronts at the terminal points.

7.1 System description and hazard identification

We can now go ahead, having in mind the steps described in section 6. As a first step, we should make a structural description of the system using a class diagram and a behavioural description by a sequence diagram. For the purposes of this paper, we can merge step 1 and step 3 of our former order, so we can directly compose the identified hazards into the diagram. Let us make some simplifications. We presume minimal failures of the electric drives and the power supply. Therefore we can consider the three potential

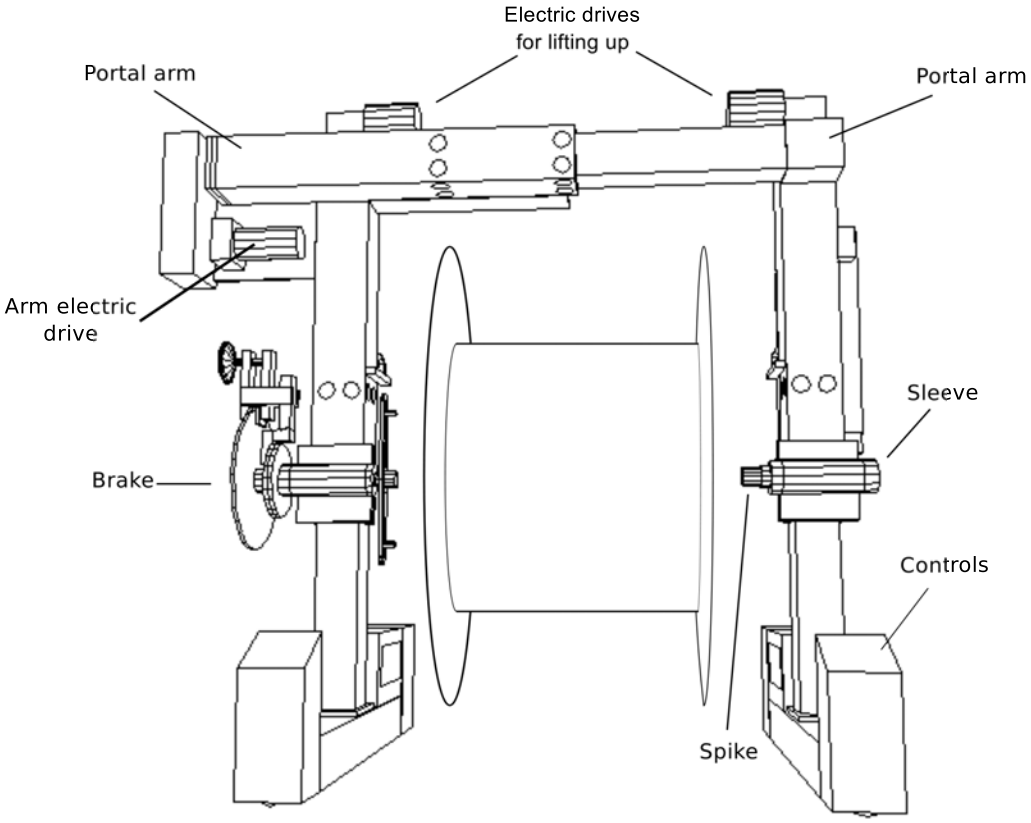

Fig. 2: Portal pay-off unit [5] with a cable reel

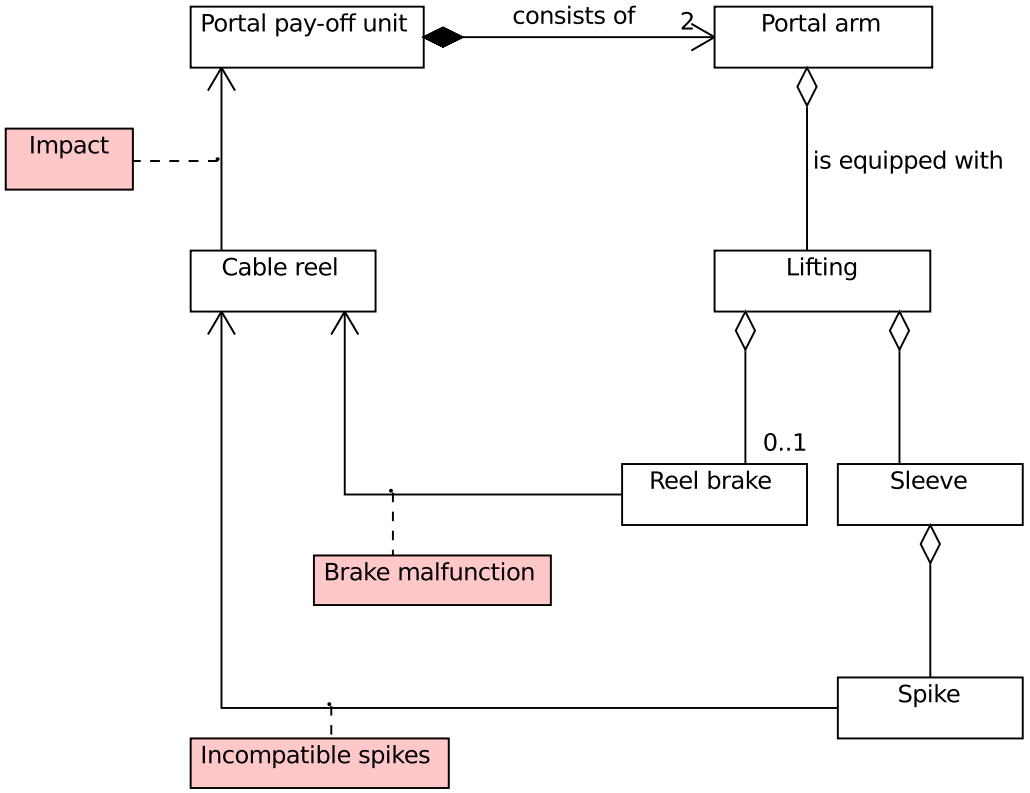

Fig. 3: Class diagram showing a pay-off unit with possible hazards

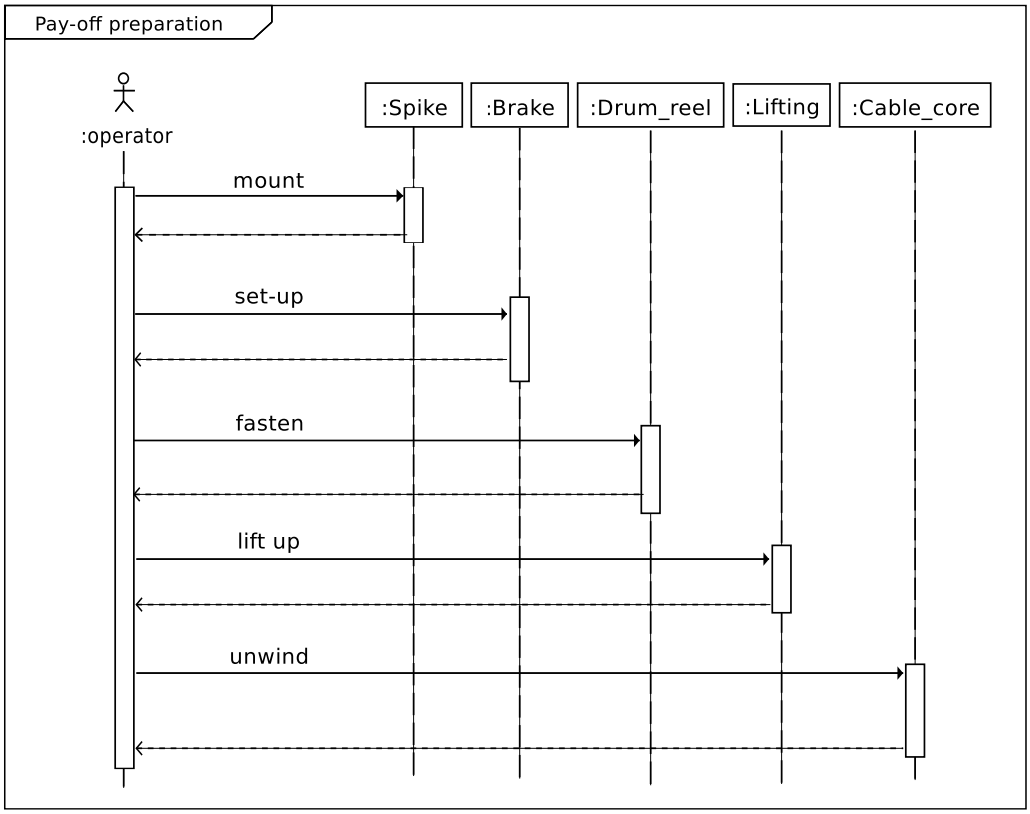

Fig. 4: Sequence diagram describing preparation for unwinding

mechanical hazards shown in Fig. 3. The class diagram includes standard UML notation [9] with rectangles as classes and junction lines as the symbols for a relationship. The lines can be equipped with short verbal phrases for better comprehension, and also with special symbols. These symbols (e.g. diamonds) represent the type of relationship and optionally the direction. It is sometimes suitable to supplement a multiplicity of specific classes. This indicates a possible count of the same classes in the relationship. The notation of multiplicity can typically be expressed in the interval form, e.g. 0..1, 2..5, where the numbers represent lower and upper bounds.

The important thing is that the diagram represents classes, not concrete objects. Hence we can reuse this diagram for other similar devices that are located in production lines.

The behavioural aspects will be presented as a preparation for the unreeling process. The sequence diagram in Fig. 4 describes the sequence of steps that an operator should take for the correct unreeling process. The rectangles with vertical dashed lines at the top of the picture are called *lifelines*, and they represent participant objects in the interaction. The thin vertical rectangles situated on the dashed lines show the activity of the specific object.

7.2 Risk scenarios

In the following step we can create approximate risk scenarios. They will be presented as Use Case diagrams with various focused objects and actors. The left part of Fig. 5 shows the first case, where the central part is the pay-off unit and external objects in the role of actors can cause possible hazards. The right side of Fig. 5 shows another situation. The central investigated object is an operator, and the external actors are the pay-off unit and the cable reel, which can threaten the operator. We can note the same graphical symbol (an icon of a "stick man" – see the UML specification in [9]) for the human actors and the technical object actors.

After the approximate scenarios, it is useful to develop a more detailed description of potential hazard occurrences, including their consequences. We can use an activity diagram for this purpose. Formally, activity diagrams arise from Petri Nets, but they differ in several ways [9]. The common attribute is the flow of a token. We will show here only the diagram for the first scenario. It is drawn in Fig. 6. Let us note that activity diagrams can, up to a point, describe actions in terms of their location. This feature can also be very useful in risk analysis. Another feature of activity diagrams is their ability to show concurrent activities and activity branching.

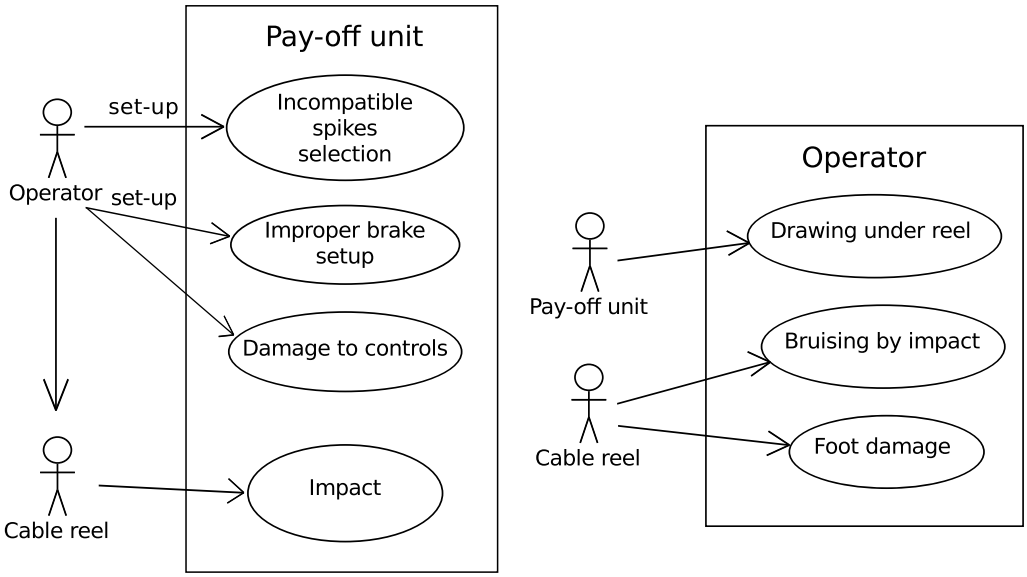

Fig. 5: The first and second scenarios as a Use Case diagram

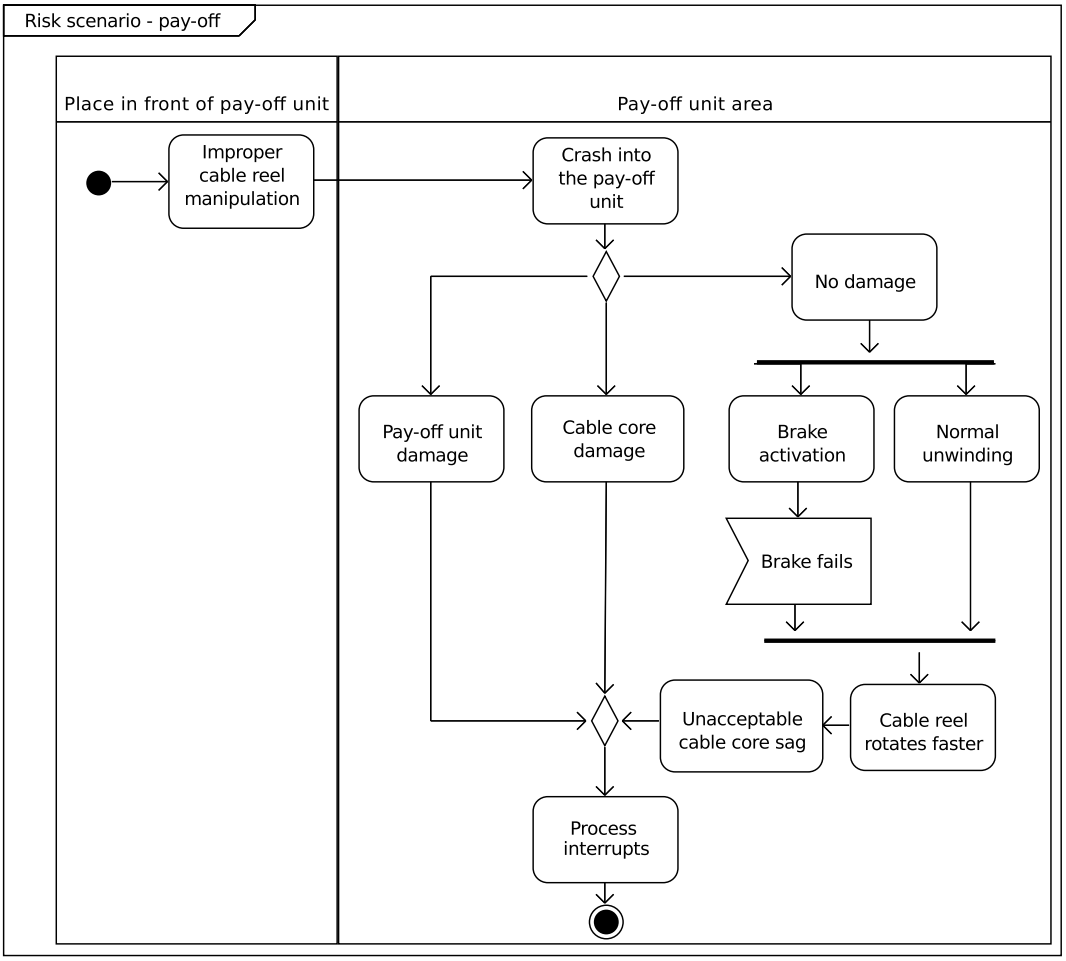

Fig. 6: The more detailed first scenario in an activity diagram

As was mentioned above, we will not make the last steps (quantification, evaluation and documentation) of our new method, which lie outside the primary goal of this paper. We have therefore completed our task.

8 Conclusion

In this paper, we have tried to present a new approach to a common interdisciplinary risk analysis process. Our aim was to present the possibilities of Unified Modelling Language as a suitable tool for risk analysis. We decided that the best way was to show it on the basis of a small practical example. The utilisation of UML is very efficient, although the tool itself comes from a different technical branch. Its abstract concept allows the modelling of complex technical and other issues, e.g. from the field of biology. It is impossible to show in a single paper the whole range of possibilities of UML, so we tried to emphasise significant structural and behavioural modelling considerations. Thanks to the origin of the Unified Modelling Language, we can easily use it to convert the model into further computer processing.

Although the application of UML seems to be very beneficial, it is also necessary to discuss the negatives. Obviously we have to learn UML, and we have to understand it. Sometimes this could be quite difficult, especially when a risk analysis is made by a whole team of experts. Another consideration is the need for a computer system. With the help of available software tools we can work better with UML-based risk models. Manipulation, storage and advanced computation of the models are much easier, but professional software tools can be very expensive. We could also draw the diagrams on paper, but in the case of huge systems this would involve a large amount of work.

Future work on this topic will involve making potential hazard templates in electro-technical manufacturing. These UML-based templates could consist of particular hazard classes and general risk scenarios, which are typical for this branch of industrial processing.

References

[1] Tichý, M.: *Risk governance: analysis and management.* 1st ed. Praha, C. H. Beck, 2006. 396 p. ISBN 80-7179-415-5. in Czech

[2] IEC 300-3-9. *Dependability management – Part 3: Application guide – Section 9: Risk analysis of technological systems.* 1995. 28 p.

[3] Arlow, J., Neustadt, I.: *UML2 and the unified process: Practical object-oriented analysis and design.* 2nd Edition. Brno, Computer Press, a.s., 2007. 567 p. ISBN 978-80-251-1503-9. in Czech

[4] IEC 60300-3-1. *Dependability management – Part 3-1: Application guide – Analysis techniques for dependability – Guide on methodology.* 2003. 55 p.

[5] *Transportkabel DIXI a.s.* [on-line]. April 2nd 2006 [cit. 2010–06–24]. Portal pay-off unit. Available from WWW: http://www.tkdixi.cz/Html_ENG/OBD2800.htm.

[6] Jirsa, J.: Methods for Analysis and Modeling of Failures in Manufacturing Systems. In *International Conference Pelincec 2005 [CD-ROM]* 2005. Warsaw, Warsaw University of Technology.

[7] IBM Rational Unified Process [on-line]. [cit. 2010–06–24] Available from WWW: http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process.

[8] FMEA RPN [on-line]. [cit. 2010–07–09] Available from WWW: http://www.fmea-fmeca.com/fmea-rpn.html.

[9] Object Management Group, Inc. *OMG Unified Modeling Language, Superstructure*, v2.1.2 edition, November 2007. [cit. 2010–07–09] Available from WWW: http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF.

Ing. Jan Jirsa
Phone: +420 224 353 965
E-mail: jirsaj@fel.cvut.cz
Department of Electrotechnology
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2, 166 27 Praha 6, Czech Republic

Doc. Ing. Jaroslav Žáček, CSc.
Phone: +420 224 352 198
E-mail: zacek@fel.cvut.cz
Department of Electrotechnology
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2, 166 27 Praha 6, Czech Republic